II 4

Atty. Dkt. No. 074273-0192

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Steven Donald SPENCE, et al.

Title: SYSTEM AND METHOD FOR CLOCK SYNCHRONIZATION FOR USB SINK DEVICE

Appl. No.: To be assigned

Filing Date: July 20, 2001

Examiner: Unassigned

Art Unit: Unassigned

CLAIM FOR CONVENTION PRIORITY

Commissioner for Patents
Washington, D.C. 20231

Sir:

The benefit of the filing date of the following prior foreign application filed in the following foreign country is hereby requested, and the right of priority provided in 35 U.S.C. § 119 is hereby claimed.

In support of this claim, filed herewith is a certified copy of said original foreign application:

- AUSTRALIA Patent Application No. PQ 8963/00 filed July 24, 2000.

Respectfully submitted,

By Phillip J. Articola Reg. No. 38,819

for David A. Blumenthal
Attorney for Applicant
Registration No. 26,257

Date July 20, 2001

FOLEY & LARDNER
Washington Harbour
3000 K Street, N.W., Suite 500
Washington, D.C. 20007-5109
Telephone: (202) 672-5407
Facsimile: (202) 672-5399

Spence
74273-192

AUSTRALIA

CERTIFIED COPY OF PRIORITY DOCUMENT

Patent Office
Canberra

I, LISA TREVERROW, TEAM LEADER EXAMINATION SUPPORT AND SALES hereby certify that annexed is a true copy of the Provisional specification in connection with Application No. PQ 8963 for a patent by NEC AUSTRALIA PTY LTD filed on 24 July 2000.

WITNESS my hand this
Twenty-ninth day of November 2000

Lisa Treverrow
LISA TREVERROW
TEAM LEADER EXAMINATION
SUPPORT AND SALES

Regulation 3.2

NEC Australia Pty Ltd

AUSTRALIA

Patents Act 1990

PROVISIONAL SPECIFICATION

for the invention entitled:

"A Clock Synchronisation Method for USB Sink Devices"

The invention is described in the following statement:

A CLOCK SYNCHRONISATION METHOD FOR USB SINK DEVICES

The present invention relates to a clock synchronisation method and a system for executing the method. In particular, the invention relates to a method for a USB (Universal Serial Bus) data sink, using an isochronous endpoint, to adaptively synchronise its output clock rate to a USB data source input clock rate, without using feed back, or feed forward techniques or additional clock signals.

In a synchronous communication system having a data source and a data sink, the data output from the data sink is expected to be synchronous with the data input to the data source. When the data source and the data sink are connected using a USB, the data input clock is not synchronous to the USB clock, which does not allow the data output to be synchronised to the data input, even if the data output clock is synchronised to the USB clock. The difference between the input and output clocks leads to a frequency mismatch and subsequent data corruption.

Some systems, which do not require a high degree of data accuracy, could tolerate the frequency mismatch and data corruption. Typical low cost audio applications using a non synchronised USB audio source (e.g. CD) and USB audio sink (e.g. speakers) are not significantly degraded audibly, if audio samples are discarded or inserted to accommodate clock mismatch.

For real-time ISDN applications such as video, data loss due to clock frequency mismatch notably affects the quality of the service. It produces effects such as picture freezing and is much less tolerable.

Techniques, which can perform clock synchronisation between the data input clock and the USB clock, have limited application due to the fact that they can only synchronise the USB clock to a limited degree of accuracy.

For an application involving a mobile phone using a USB interface to access synchronous services via an ISDN interface, one possible synchronisation option is a method referred to as clock mastering for asynchronous source and synchronous sink. This involves the source device influencing a USB host's SOF (Start Of Frame) generation, so that isochronous data transfer is synchronised to the source. The host SOF rate (1mS) is adjusted to track the mobile phone's data frame rate (10mS). An ISDN interface is a USB synchronous sink (i.e. locked to the SOF clock) and the frame transfer from the mobile phone to the ISDN interface is synchronised. The ISDN clock (i.e. I.430 192KHz bit clock) is synchronised to the SOF clock. The SOF adjustment resolution is 1/12000 bit times, where a bit period is one 12 MHz clock cycle or 83 ppm. Due to this coarse clock resolution, the source is not able to select a single optimum SOF period. The source clock of a mobile phone is locked to the mobile network and is very accurate with the result that the frequency error may approach 83 ppm. As this would still cause frame drift, so the source needs to continuously switch the SOF bit period up and down to achieve a synchronised average SOF period. The sink must then track the average of the SOF period, as a step of 83 ppm (parts per million) is not acceptable. The I.430 interface imposes a clock accuracy requirement of better than ±100 ppm and clock jitter must be significantly less. To overcome this it is necessary to integrate the frequency change with a period greater than the correction rate, so that the applied sink frequency correction is much more stable. An error of 83 ppm causes an I.430 frame slip approximately every 3 seconds, so a correction rate of 3 Hz and an integration period of 3 seconds is a possible solution.

Whilst clock mastering is feasible, it is less than desirable due to the coarse clock adjustment and the limitation that only one device may act as clock master. In a number of applications this synchronisation method could not be used.

It is desired to provide another synchronisation option, which involves an asynchronous source and adaptive sink, and allows implicit feed forward clock recovery, or which at least provides a useful alternative.

In accordance with the present invention, there is provided a method for adaptive synchronisation of a data sink to a data source coupled by a USB, including the steps of:

receiving data for a buffer of said sink at an average data rate representative of the data rate of said source;

determining a data level for said buffer based on input packet size and output packet size;

comparing an accumulated data level for said buffer with a threshold level; and

correcting a clock frequency for said sink when said accumulated data level exceeds said threshold level.

Preferably said correcting step involves correcting the frequency by an amount equal to a constant K divided by the time required for the accumulated data level to drift from a reference level to the threshold level.

Preferably said method includes inhibiting said comparing and correcting step for a predetermined period after said correcting step. The predetermined period may be between three or five times said drift time. Preferably said predetermined period is reduced if said data level traverses said reference level or exceeds twice the threshold level.

Preferably the reference level is the data level measured over a first measurement period. Preferably said comparing step is executed periodically.

Preferably the threshold level is set to be greater than three times a maximum data level jitter. Preferably the size of the buffer is set to be greater than three times said threshold level.

Preferred embodiments of the present invention are hereinafter described, by way of example only, with reference to the accompanying drawings, wherein:

Figure 1 is a block diagram of a preferred embodiment of a data transfer system excecuting an adaptive clock synchronisation method;

Figure 2 is a graph of the level of data in a sink device of the system against time; and Figure 3 is a graph of the average sink data level against time.

A data transfer system, as shown in Figure 1, executes adaptive clock synchronisation for data passed from a source device to a sink device using a USB, and a USB host that executes a USB host application. The host may be a PC with at least one USB port. The source device may be a mobile telephone and the sink device may be a computer with a video card or an interface to an ISDN, as described below. The source has a synchronous data flow rate ***DRsource***, a source frequency ***Fsource***. The USB source to host average data flow rate is ***DRusb1***, and the USB host to sink average data flow rate is ***DRusb2***. The host application preserves the data flow rate from the source connection to the sink connection such that ***DRusb1*** = ***DRusb2*** = ***DRusb***. The sink data flow rate is denoted ***DRsink*** and the sink frequency is denoted ***Fsink***. The sink device includes a rate adaption buffer which has an input stream received at ***DRusb1*** and an output stream sent at the rate ***DRsink***. The sink device includes circuit components to implement the buffer and execute the synchronisation method. For instance, the components may include a microprocessor and control software defining the synchronisation method.

The adaptive clock synchronisation method, as described in detail below, operates by adjusting the sink data rate ***DRsink*** to match the USB data delivery rate ***Drusb*** by controlling the sink clock frequency ***Fsink***. This results in synchronisation of the sink device with the source as the source data rate ***DRsource*** is implicit in the USB data delivery rate ***DRusb*** when averaged over time.

The difference between the input and output data streams of the rate adaption buffer is detected by comparing the sink device data level (being input flow from the USB minus the output flow), with plus and minus threshold levels. The data level is checked periodically by the sink device and if the threshold is exceeded, then a frequency correction is determined based on the inverse of the time (***Tdrift***), taken for the data level to drift from a reference level to the threshold level, as shown in Figure 3. The correction is set approximately 20% greater than necessary, to ensure the drift is reversed and the frequency asymptotes towards the source frequency. Recorrection is inhibited for 5 times the drift period (***Tdrift***) or terminated early, if the data level crosses the reference level or 2 times the threshold level. These checks cater for error conditions should they arise. The reference level is determined when data transfer commences.

The flow rate matching process results in frequency matching (***Fsource*** to ***Fsink***) to better than 100 ppm and typically better than 10ppm (this meets the requirements of BRI ISDN) and an elimination of data overrun or underrun, which occurs when source and sink data rates are not matched.

Figure 2 shows how the level of data held in the rate adaption buffer of the sink device varies over time. In this example data is shown arriving in small roughly constant packet sizes (***Pi***) and being extracted at a fixed rate with a larger packet size (***Po***). The average data level is an important factor for determining flow imbalance. The data level increase is greatly exaggerated for the purposes of this description. For an ISDN interface, a rate difference of 50 ppm will cause the level to change by 1 byte over about 1 second.

Figure 3 shows typical behaviour of the average data transfer level. Two clock correction points are shown. The rate of drift is slowed at each correction. The overshoot shown at the correction point occurs due to the correction being applied gradually.

To execute the clock synchronisation method the system has:

1. A data source with a synchronous data flow rate, ***DRsource***.
2. ***DRusb*** = ***DRsource***, when measured over sufficiently long period, to average out the transport jitter.
3. All source data is transferred to the USB host. This may involve data repacketisation and data packets (***Pi***) with different size, but constant in time, between the USB frames.
4. A USB transport rate which is greater than the source data rate. The source data

delivery rate, ***DRusb*** is not the same as the USB transport rate. ***DRusb*** is the rate at which the USB system transports data provided by the source.

5. A host application which preserves the data flow rate from the source connection to the sink connection, such that ***DRusb1*** = ***DRusb2*** = ***DRusb***, as described above. Figure 1 shows the source and sink both connected to the host by a USB, yet all that is required is the sink to receive data at a data delivery rate ***DRusb2*** that implies the source rate ***DRsource***.

The USB system introduces data transfer jitter (ie packet size (***Pi***) variation) due to the phase shift between the source clock and the USB clock, and consequently output packet size (***Po***) variation. The adaptive synchronisation method takes this into account, by buffering the data transfer and checking only for long term accumulation or depletion of data, passing through the sink, to determine data rate imbalance.

The data level variation over time, as shown in Figure 2, is a result of the combined effect of the following factors:

(i) ***Fsink*** to ***Fsource*** differences at start up (i.e. prior to clock correction);
(ii) USB transport jitter; and
(iii) Sink frequency jitter (arising from the synchronisation method).

As long as the rate adaption buffer is large enough to accommodate the fluctuation (i.e. does not overflow or underflow) the synchronous output data flow from the sink can be maintained uninterrupted. Balanced data flow requires the quantity of input data minus the quantity of output data to remain constant over time, such that:

(a) If the data level increases then ***Fsource*** > ***Fsink*** and ***Fsink*** must be increased; and
(b) If the data level decreases then ***Fsink*** > ***Fsource*** and ***Fsink*** must be decreased.

Data flow balance in the reverse direction occurs simultaneously as a result of synchronisation between transmit and receive clocks in both the source and sink.

As shown in Figure 2, the data level changes constantly. In order to determine if there is a flow imbalance (and hence a clock frequency difference), it is necessary to determine a trend in the data level changes. This is done by accumulating the data level changes, i.e. the sum of the data input flow minus the data output flow. The data input flow is the total size of the packets received by the buffer and the data output flow is the total size of the packets output by the buffer.

The Accumulated Data Level (*ADL*) over a period of time (*ΔT*) is:

$$\boldsymbol{ADL = Reference\_level + \sum_{0}^{\Delta T}(Data\_input - Data\_output)} \qquad (1)$$

i.e. $\boldsymbol{ADL = Reference\_level + \Delta DL}$

where

Reference_level	is the reference (or initial) data level
Data_input	is the data input flow, being $\sum P_i$
Data_output	is the data output flow, being $\sum P_o$
ΔDL	is the accumulated data level change

Only changes in the accumulated data level are of interest for clock correction , because the reference data level is constant.

The accumulated data level (***ΔDL***) varies due to:

(i) The frequency difference between the source and sink; and

(ii) Transport jitter due to the discrete nature of data inflow and outflow.

Thus the change in the Accumulated Data level can be defined also as,

$$\boldsymbol{\Delta DL = \sum(Freq\_difference\_level\_drift + Packet\_del/extr\_jitter)}$$

(2)

where

Freq_difference_level_drift is the data level change
Packet_del/extr_jitter is the jitter due to the source and sink data flow

The second factor, the jitter, causes an error in the measurement of the data level and in the calculation of the frequency correction required to achieve balance. To minimise the jitter influence, the USB data transfer packet size should be kept as small as possible and the data level measurement synchronised to the source or sink interface which delivers/retrieves data to/from the interface with greater packet size (***Pi*** or ***Po***).

Hereafter, the adaptive clock synchronisation method is described with reference to the USB sink being a BRI ISDN adaptor or interface and the USB source being a mobile phone. The mobile phone receives input data, such as a video data, in 10mS frames (80 bytes of data). When transported over the USB (as source data), the data frame is spread over nine 1mS isochronous USB frames (nominally 11 byte frames) and read by the sink device as 10mS frames (80 bytes). The data level measurement is synchronised to the data retrieval at the sink, resulting in a jitter of 11 bytes (a USB source packet). If the data level measurement is not synchronised, the potential jitter is 91 bytes (a USB source packet + a sink packet).

The first factor causing variations in the accumulated data level (*ΔDL*) is due to the source - sink frequency difference. The relationship of the ***Freq_difference_level_drift*** to the frequency difference is derived as follows:

$$\Delta DL \propto Fsource - Fsink$$

$$\Delta DL = K \times \int_{0}^{\Delta T} (Fsource - Fsink)\, dt$$

$$\Delta DL = K \times \Delta T \times (Fsource - Fsink)$$

(3)

where

ΔT is the measurement period

K is a system design constant

A frequency correction value (***Fcorr***) is equal to the frequency difference between the source and the sink frequencies (***Fsource – Fsink***) and is determined by rearranging Equation 3, setting the data level to a suitable error detection threshold and setting ΔT to the time for the data level to drift from the reference level to the positive or negative threshold level, as shown in Figure 3, such as:

$$\textit{Fcorr} = \textit{Threshold\_level} \,/\, (K \times \textit{Tdrift}) \tag{4}$$

The frequency correction value is therefore determined by measuring the time taken for the data level to shift from the reference level to a threshold level. The key factor in doing this is to accurately determine the data level and its change. The method and an appropriate set of parameters for a given USB stream and an ISDN interface are described below.

The threshold level is the level which represents a significant/detectable change of the average buffer level. It should be:

(i) As low as possible to minimise the rate adaption buffer size; and

(ii) As high as possible to reduce the impact of jitter error on frequency correction accuracy.

Jitter may cause the threshold level to be exceeded earlier and the frequency correction value to be incorrectly calculated. A small ***Tdrift*** measure produces an overcorrection, as the correction factor is inversely proportional (as shown in Equation 4). This is acceptable as long as the flow drift is reversed and the magnitude of the frequency error reduced, so that successive corrections asymptote the error to zero. The level of overcorrection only impacts on the rate of convergence of the frequency error and does not effect data transfer quality.

Hence a maximum overcorrection due to jitter of 50% is recommended. This is comfortably below the 100% limit over which the sink frequency will diverge from the source frequency.

The threshold level, so that jitter does not cause an overcorrection greater than 50%, is determined from Equation 4 as follows:

Fcorr = Threshold_level / (K x Tdrift)	without jitter
Fcorr_j = Threshold_level / (K x Tdrift_j)	with jitter

For a 50% overcorrection, i.e. ***Fcorr_j / Fcorr*** = 1.5
then ***Tdrift / Tdrift_j*** = 1.5 or ***Tdrift_j*** = 2/3 ***Tdrift***

This occurs when the jitter level is 30% of the threshold level. Hence the threshold level must be at least 3 times the jitter level.

The data level is periodically compared (at each monitoring period) with the threshold level to minimise processing requirements.

For the ISDN example the USB packet size is nominally 11 bytes, the maximum jitter error is 11 bytes, the threshold level = 3 x 11 = 33 bytes, and the monitoring period = 1 second.

The data level change due to a maximum frequency error of 100 ppm over 1 second = 1.25 bytes (based on Equation 5). The correction response is not effected by a one second monitoring period.

Clock correction is undertaken when the accumulated data level change (ΔDL) becomes equal or exceeds (in absolute terms) the threshold level. A frequency correction value ***Fcorr*** is determined, the sink clock ***Fsink*** is adjusted by ***Fcorr*** and further correction inhibited for a period to allow the data level to asymptote back towards the reference level.

The correction value ***Fcorr*** is increased by 20 % over that derived from Equation 4 to accommodate calculation error and ensure the flow imbalance is reversed. The overcorrection factor is implementation dependent. It must exceed any calculation error resulting from the implementation but should not exceed 100% minus the overcorrection which results due to data level jitter. The synchronisation method asymptotes the sink frequency ***Fsink*** to a value, which slowly hunts between a slightly high and slightly low frequency about the actual source frequency ***Fsource***. This is also done to accommodate the limitations in clock setting resolution. The method operates through the use of the ± threshold levels and a minimum correction value.

For the ISDN example, the minimum adjustment is ±1ppm. Note also the maximum correction is also limited to ±100 ppm to comply with I.430 requirements.

The sink clock frequency adjustment rate is limited to provide an acceptable output clock jitter. For the ISDN example the adjustment rate is restricted to 1ppm per 10mS.

The correction inhibit period is employed to give time for the correction to take effect. This is needed because:

(a) Jitter will cause the data level value to recross the threshold until the average data level fails below the threshold minus the maximum jitter;

(b) The limited clock adjustment rate may cause the data level to increase further before the full correction takes place; and

(c) The flow imbalance, though reversed by the correction, is reduced in magnitude and may require considerable time for the data level to reduce from the threshold level to the reference level.

The inhibit period is set to between 3x and 5x ***Tdrift*** and terminated immediately if the reference level or twice the threshold level is crossed. The factor of 5 results from the frequency correction being nominally overcorrected by 20%. Thus the time for the data level to reduce back to the reference level is ***Tdrift***/0.2. The factor may be as low as 3x as this will still ensure the data level has decreased to a level less than 1/2 the threshold and jitter will not cause false recorrection. The termination conditions are included to accommodate the additional overcorrection due to jitter and further change of the source frequency ***Fsource***.

The factor ***K*** in Equation 3 is determined by setting ***ΔT***, (***Fsource – Fsink***) and ***ΔDL*** to known values for the application. For a 64 Kbps data stream, 1 byte of data is transmitted every 125µS. If the frequency error (***Fsource – Fsink***) is 1ppm, the data level will increase by 1 byte in 125µS/10^{-6} = 125 seconds. So by setting ***ΔT*** = 125 in Equation 3,

$$\boldsymbol{K = \Delta DL / (\Delta T \text{ x } (Fsource - Fsink))} = 1/(125 \text{ x } 1) = 1/125 \text{ s}^{-1} \text{ ppm}^{-1}$$

Equation 4 becomes:

$$\boldsymbol{Fcorr} = 125.\text{x } \boldsymbol{Threshold / Tdrift} \quad \text{ppm} \qquad (5)$$

where ***Tdrift*** has units of seconds.

For the ISDN example the sink clock is implemented as a VCO controlled by a DAC output from a microprocessor that monitors the rate adaption buffer and executes the synchronisation method to determine ***Fcorr***. The VCO has a ±2V input range to adjust the frequency ±100 ppm. So for a 5V DAC range, 1 bit adjusts the frequency 0.98 ppm. Including the 20% overcorrection, the constant ***K*** becomes 153.6 (= 125 x 1.2 / 0.98) and the DAC output correction value (***DACout***)

$$\boldsymbol{DACout} = 153.6 \text{ x } \boldsymbol{Threshold / Tdrift} \qquad \text{bits/ppm}$$

Many modifications will be apparent to those skilled in the art without departing from the scope of the present invention as hereinbefore described with reference to the accompanying drawings.

DATED this 24th day of July 2000

NEC Australia Pty Ltd
By its Patent Attorneys
DAVIES COLLISON CAVE

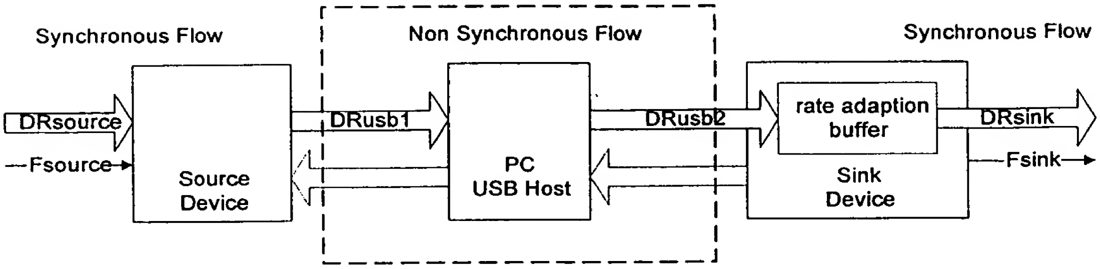

Figure 1

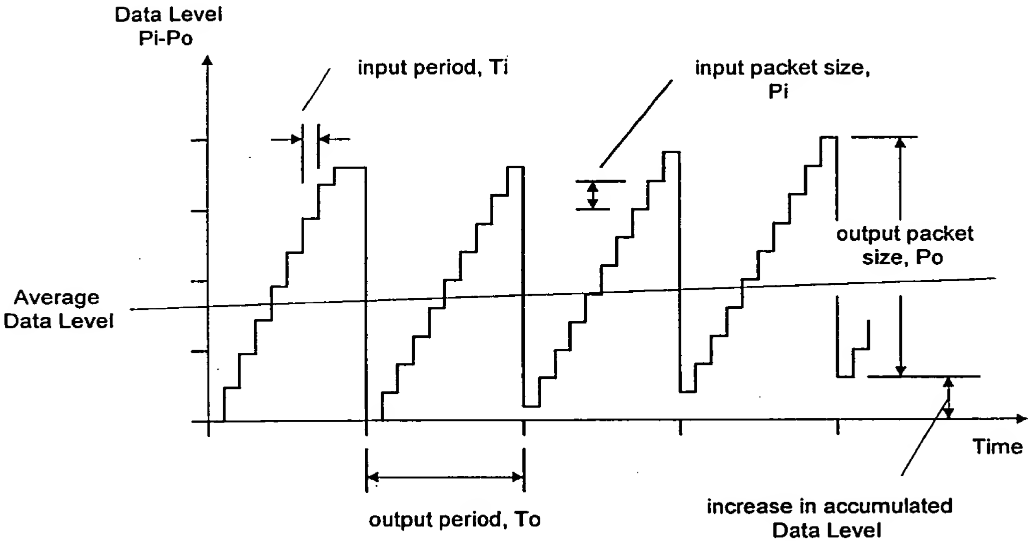

Figure 2